**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

**Отчет о лабораторной работе № 10**

по теме

«Java DataBase Connectivity»

**Выполнили:**

Студенты гр. 045

Бригады №10

Анохин В.А.

Вашкулатов Н.А.

**Проверил:**

Асс. Бастрычкин А.С.

Цель работы:изучение взаимодействия языка программирования Java с базами данных, приобретение навыков работы с БД в Java-программах.

Ход работы

Задание

Города. В БД хранится информация о городах и их жителях.

Для городов необходимо хранить:

• название;

• год основания;

• площадь;

• количество населения для каждого типа жителей.

Для типов жителей необходимо хранить:

• город проживания;

• название;

• язык общения.

• Вывести информацию обо всех жителях заданного города, разговаривающих на заданном языке.

• Вывести информацию обо всех городах, в которых проживают жители выбранного типа.

• Вывести информацию о городе с заданным количеством населения и всех типах жителей, в нем проживающих.

Код программы:

Класс ConnectionCreator:

import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
  
public class ConnectionCreator {  
 public static Connection getNewConnection(String username, String password, String url) throws SQLException {  
 return DriverManager.getConnection(url,username,password);  
 }  
}

Класс DBAction:

import org.example.models.City;  
import org.example.models.Type;  
import java.sql.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DBAction{  
 final private static String USERNAME = "root";  
 final private static String PASSWORD = "root";  
 final private static String URL = "jdbc:mysql://localhost:3306/lab10";  
 final private static String SELECT\_FROM\_CITY = "select \* from city";  
 final private static String SELECT\_FROM\_TYPE = "select \* from type";  
 final private static String FIRST\_QUERY = "select c.name, t.name, t.language, ct.value " +  
 "from city c join city\_type ct on c.id = ct.city\_id " +  
 "join type t on ct.type\_id = t.id " +  
 "where c.name = ? and t.language = ? ";  
 final private static String SECOND\_QUERY = "select c.\*, t.name, ct.value " +  
 "from city c join city\_type ct on c.id = ct.city\_id " +  
 "join type t on ct.type\_id = t.id " +  
 "where t.name = ? ";  
 final private static String THIRD\_QUERY = "select c.\*, t.\*, ct.value \n" +  
 " from city c join city\_type ct on c.id = ct.city\_id\n" +  
 " join type t on ct.type\_id = t.id\n" +  
 " where c.id in (select c.id\n" +  
 " from city c join city\_type ct on c.id = ct.city\_id\n" +  
 " group by c.id\n" +  
 " having sum(ct.value) = ?)";  
   
 public static void printAllCities() {  
 List<City> cities = new ArrayList<>();  
 try {  
 Connection connection = ConnectionCreator.getNewConnection(USERNAME, PASSWORD, URL);  
 Statement statement = connection.createStatement();  
 try {  
 ResultSet resultSet = statement.executeQuery(SELECT\_FROM\_CITY);  
 while (resultSet.next()) {  
 City city = new City();  
 city.setId(resultSet.getInt("id"));  
 city.setName(resultSet.getString("name"));  
 city.setYear(resultSet.getInt("year"));  
 city.setSquare(resultSet.getInt("square"));  
  
 cities.add(city);  
 }  
 System.out.println();  
 System.out.println("Все города: ");  
 cities.forEach(System.out::println);  
 resultSet.close();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 System.err.println("Ошибка выполнения запроса");  
 }finally {  
 connection.close();  
 statement.close();  
 }  
  
 } catch (SQLException e) {  
 e.printStackTrace();  
 System.err.println("Ошибка подключения");  
 }  
 }  
  
 public static void printAllTypes(){  
 List<Type> types = new ArrayList<>();  
 try {  
 Connection connection = ConnectionCreator.getNewConnection(USERNAME, PASSWORD, URL);  
 Statement statement = connection.createStatement();  
 try {  
 ResultSet resultSet = statement.executeQuery(SELECT\_FROM\_TYPE);  
 while (resultSet.next()){  
 Type type = new Type();  
 type.setId(resultSet.getInt("id"));  
 type.setName(resultSet.getString("name"));  
 type.setLanguage(resultSet.getString("language"));  
 types.add(type);  
 }  
 System.out.println();  
 System.out.println(" Все типы жителей: ");  
 types.forEach(System.out::println);  
 resultSet.close();  
 } catch (SQLException e){  
 e.printStackTrace();  
 System.err.println("Ошибка выполнения запроса");  
 } finally {  
 connection.close();  
 statement.close();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 System.err.println("Ошибка подключения");  
 }  
 }  
  
 public static void printFirstQuery(String city, String language){  
 try {  
 Connection connection = ConnectionCreator.getNewConnection(USERNAME, PASSWORD, URL);  
 PreparedStatement preparedStatement = connection.prepareStatement(FIRST\_QUERY);  
 try {  
 preparedStatement.setString(1,city);  
 preparedStatement.setString(2,language);  
 ResultSet resultSet = preparedStatement.executeQuery();  
 while (resultSet.next()){  
 System.out.println("[" + resultSet.getString("c.name") + " " +  
 resultSet.getString("t.name") + " " +  
 resultSet.getString("t.language") + " " +  
 resultSet.getString("ct.value") + "]");  
 }  
 resultSet.close();  
 } catch (SQLException e){  
 e.printStackTrace();  
 System.err.println("Ошибка выполнения запроса");  
 }finally {  
 connection.close();  
 preparedStatement.close();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 System.err.println("Ошибка подключения");  
 }  
 }  
  
 public static void printSecondQuery(String type){  
 try {  
 Connection connection = ConnectionCreator.getNewConnection(USERNAME, PASSWORD, URL);  
 PreparedStatement preparedStatement = connection.prepareStatement(SECOND\_QUERY);  
 try {  
 preparedStatement.setString(1,type);  
 ResultSet resultSet = preparedStatement.executeQuery();  
 while (resultSet.next()){  
 System.out.println("[" + resultSet.getString("c.name") + " " +  
 resultSet.getInt("c.year") + " " +  
 resultSet.getInt("c.square") + " " +  
 resultSet.getString("t.name") + " "+  
 resultSet.getString("ct.value") + "]");  
 }  
 resultSet.close();  
 } catch (SQLException e){  
 e.printStackTrace();  
 System.err.println("Ошибка выполнения запроса");  
 }finally {  
 connection.close();  
 preparedStatement.close();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 System.err.println("Ошибка подключения");  
 }  
 }  
  
 public static void printThirdQuery(String value){  
 try {  
 Connection connection = ConnectionCreator.getNewConnection(USERNAME, PASSWORD, URL);  
 PreparedStatement preparedStatement = connection.prepareStatement(THIRD\_QUERY);  
 try {  
 preparedStatement.setString(1,value);  
 ResultSet resultSet = preparedStatement.executeQuery();  
 while (resultSet.next()){  
 System.out.println("[" + resultSet.getString("c.name") + " " +  
 resultSet.getInt("c.year") + " " +  
 resultSet.getInt("c.square") + " " +  
 resultSet.getString("t.name") + " "+  
 resultSet.getString("t.language") + " "+  
 resultSet.getString("ct.value") + "]");  
 }  
 resultSet.close();  
 } catch (SQLException e){  
 e.printStackTrace();  
 System.err.println("Ошибка выполнения запроса");  
 }finally {  
 connection.close();  
 preparedStatement.close();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 System.err.println("Ошибка подключения");  
 }  
 }  
}

**Основная программа**:

import java.util.Scanner;  
  
public class App  
{  
 public static void main( String[] args )  
 {  
 final String USERNAME = "root";  
 final String PASSWORD = "root";  
 final String URL = "jdbc:mysql://localhost:3306/lab10";  
 DBAction.printAllCities();  
 DBAction.printAllTypes();  
 Scanner sc = new Scanner(System.in);  
 System.out.println("Введите название города");  
 String city = sc.nextLine();  
 System.out.println("Введите название языка");  
 String language = sc.nextLine();  
 System.out.println("Вывести информацию обо" +  
 " всех жителях заданного города, " +  
 "разговаривающих на заданном языке:");  
 DBAction.printFirstQuery(city, language);  
 System.out.println("Вывести информацию обо всех городах," +  
 " в которых проживают жители выбранного типа: ");  
 System.out.println("Введите название типа");  
 String type = sc.nextLine();  
 DBAction.printSecondQuery(type);  
 System.out.println();  
 System.out.println("Вывести информацию о городе с заданным количеством населения" +  
 " и всех типах жителей, в нем проживающих: ");  
 System.out.println("Введите количество людей");  
 Integer value = sc.nextInt();  
  
 DBAction.printThirdQuery(value.toString());  
 sc.close();  
 }  
}

Создание базы данных:

CREATE DATABASE lab10

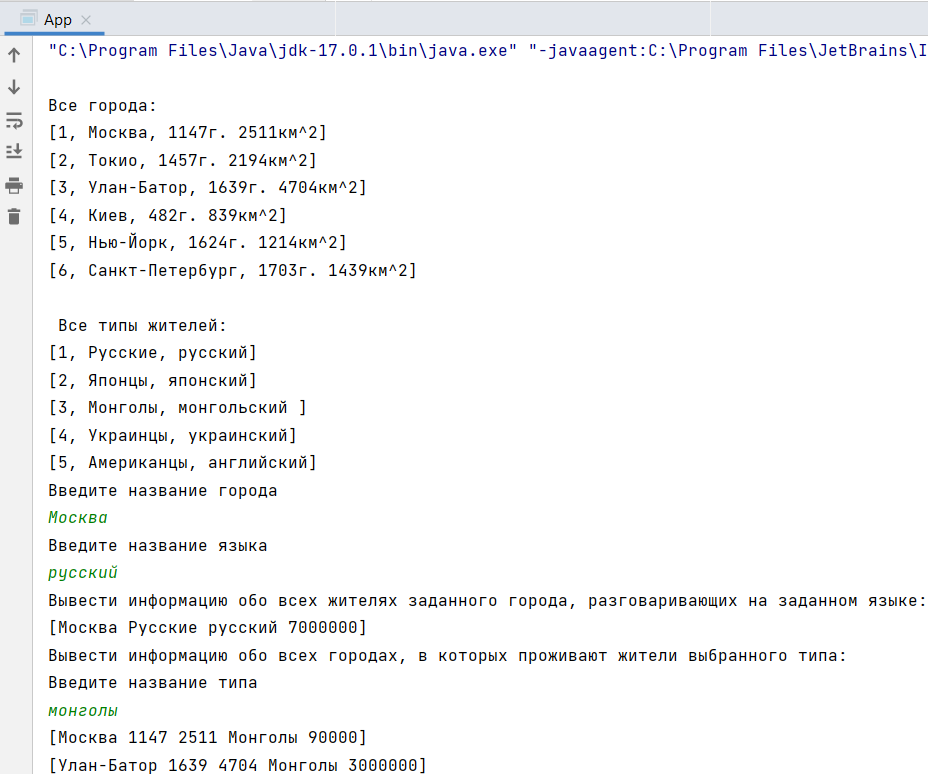
Создание таблиц:

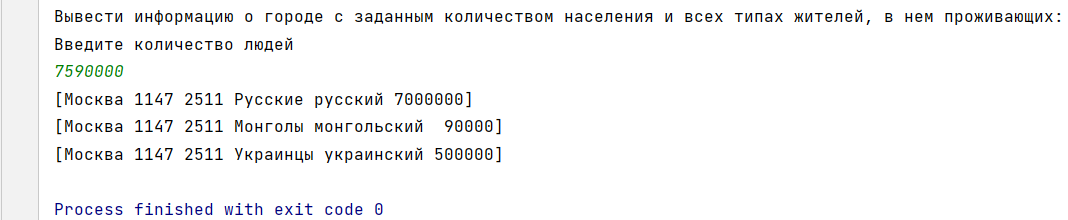
create table city  
(  
 id int auto\_increment,  
 name varchar(50) not null,  
 year int null,  
 square int null,  
 constraint city\_pk  
 primary key (id)  
);  
  
create table type  
(  
 id int auto\_increment,  
 name varchar(50) not null,  
 language varchar(50) not null,  
 constraint type\_pk  
 primary key (id)  
);  
  
create table city\_type  
(  
 city\_id int not null,  
 type\_id int not null,  
 value int,  
 constraint city\_type\_pk  
 primary key (city\_id, type\_id),  
 constraint city\_type\_city\_id\_fk  
 foreign key (city\_id) references city (id),  
 constraint city\_type\_type\_id\_fk  
 foreign key (type\_id) references type (id)  
);

Заполнение базы данных:

insert into city (name, year, square) values ("Москва", 1147, 2511);  
insert into city (name, year, square) values ("Токио", 1457, 2194); insert into city (name, year, square) values("Улан-Батор",1639,4704);  
insert into city (name, year, square) values ("Киев", 482, 839); insert into city (name, year, square) values ("Нью-Йорк", 1624, 1214);  
insert into city (name, year, square) values ("Санкт-Петербург", 1703, 1439);  
  
insert into type (name, language) values ("Русские", "русский");  
insert into type (name, language) values ("Японцы", "японский");  
insert into type (name, language) values ("Монголы", "монгольский ");  
insert into type (name, language) values ("Украинцы", "украинский");  
insert into type (name, language) values ("Американцы", "английский");  
  
insert into city\_type values (1,1,7000000);  
insert into city\_type values (1,3,90000);  
insert into city\_type values (1,4,500000);  
insert into city\_type values (2,2,5000000);  
insert into city\_type values (2,5,400000);  
insert into city\_type values (3,3,3000000);  
insert into city\_type values (3,1,100000);  
insert into city\_type values (4,4,2000000);  
insert into city\_type values (4,1,500000);  
insert into city\_type values (5,5,5000000);  
insert into city\_type values (5,1,100000);  
insert into city\_type values (6,1,5000000);  
insert into city\_type values (6,4,400000);

Окно вывода:





**Вывод:** мы изучили взаимодействия языка программирования Java с базами данных, приобрели навыки работы с БД в Java-программах.